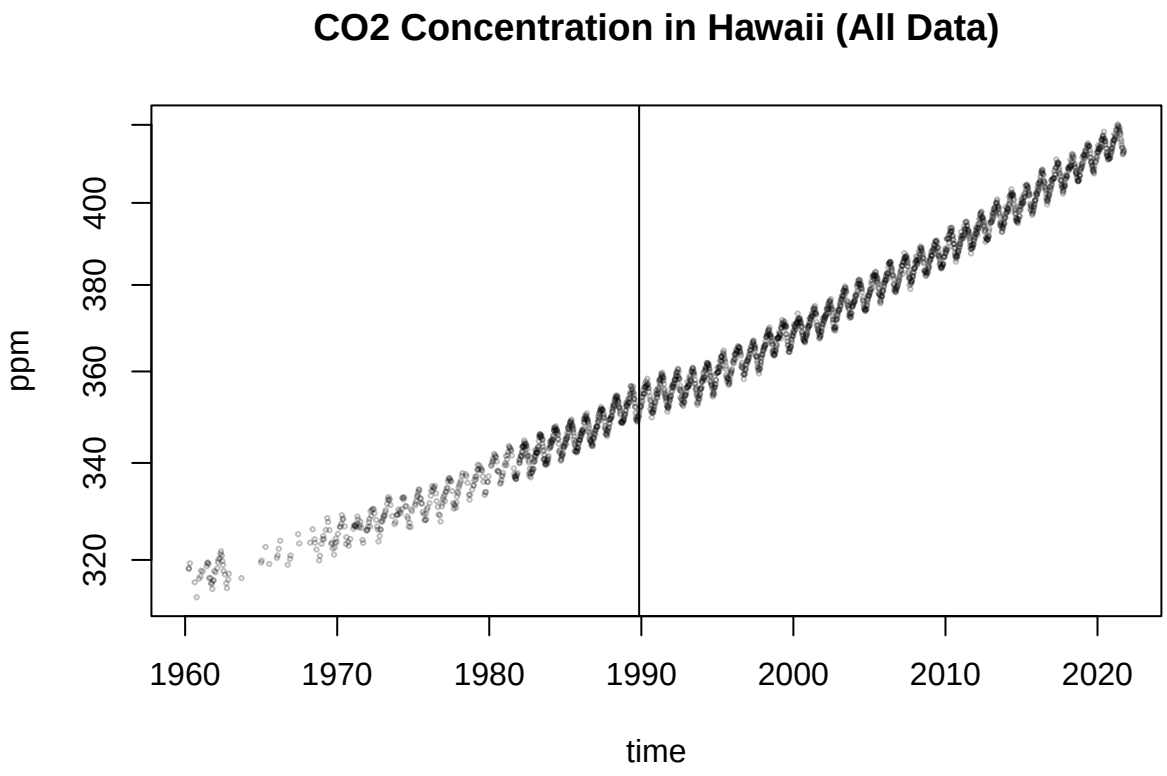
Appendix

STA442 Homework 2

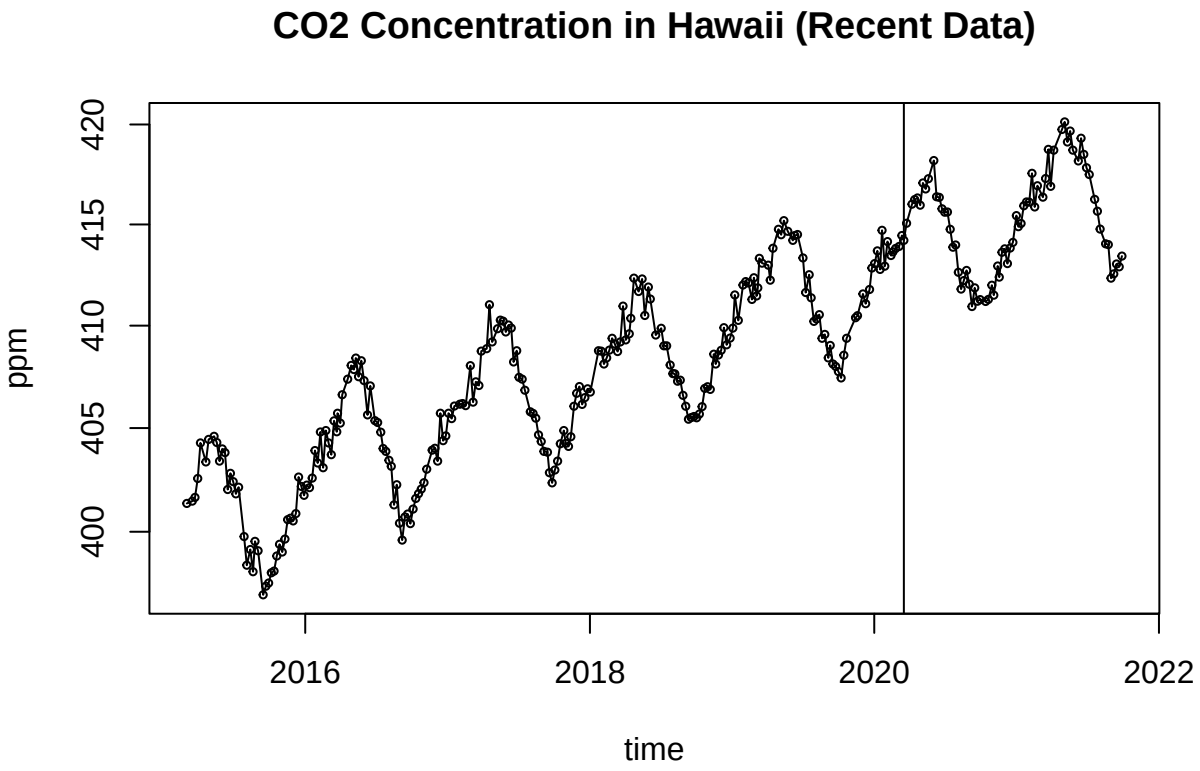
Jiaqi Bi

```
cUrl = paste0("http://scrippsco2.ucsd.edu/assets/data/atmospheric/",
  "stations/flask_co2/daily/daily_flask_co2_mlo.csv")
cFile = basename(cUrl)
if (!file.exists(cFile)) download.file(cUrl, cFile)
co2s = read.table(cFile, header = FALSE, sep = ",",
  skip = 69, stringsAsFactors = FALSE, col.names = c("day",
    "time", "junk1", "junk2", "Nflasks", "quality",
    "co2"))
co2s$date = as.Date(co2s$day)
co2s$time = strptime(paste(co2s$day, co2s$time), format = "%Y-%m-%d %H:%M",
tz = "UTC")
# remove low-quality measurements
co2s = co2s[co2s$quality == 0, ]
```

```
plot(co2s$date, co2s$co2, log = "y", cex = 0.3, col = "#00000040",
  xlab = "time", ylab = "ppm", main="CO2 Concentration in Hawaii (All Data)")
abline(v=as.Date("1989/11/09"))
```



```
plot(co2s[co2s$date > as.Date("2015/3/1"), c("date",
  "co2")], log = "y", type = "o", xlab = "time",
  ylab = "ppm", cex = 0.5, main="CO2 Concentration in Hawaii (Recent Data)")
abline(v=as.Date("2020/03/17"))
```




```
co2s$dateWeek = as.Date(lubridate::floor_date(co2s$date,
  unit = "week"))
co2s$timeYears = as.numeric(co2s$date)/365.25
co2s$cos12 = cos(2 * pi * co2s$timeYears)
co2s$sin12 = sin(2 * pi * co2s$timeYears)
co2s$cos6 = cos(2 * 2 * pi * co2s$timeYears)
co2s$sin6 = sin(2 * 2 * pi * co2s$timeYears)
allDays = seq(from = min(co2s$dateWeek), to = as.Date("2030/1/1"),
  by = "7 days")
table(co2s$dateWeek %in% allDays)
```

```
##
## TRUE
## 2240
```

```
co2s$dateWeekInt = as.integer(co2s$dateWeek)

library("INLA", verbose = FALSE)
```

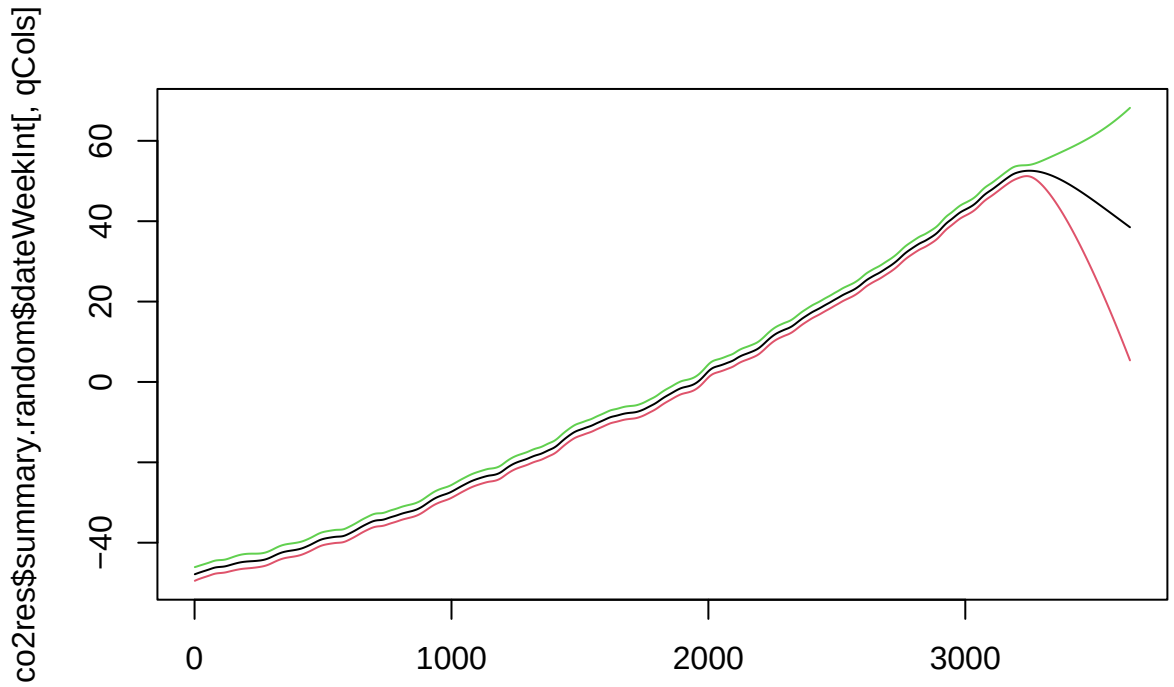
```
## Loading required package: Matrix
```

```
## Loading required package: foreach

## Loading required package: parallel

## Loading required package: sp

## This is INLA_21.02.23 built 2021-10-07 01:02:09 UTC.
##  - See www.r-inla.org/contact-us for how to get help.
##  - To enable PARDISO sparse library; see inla.pardiso()
##  - Save 350.5Mb of storage running 'inla.prune()'
```

```
# disable some error checking in INLA
mm = get("inla.models", INLA:::inla.get.inlaEnv())
if (class(mm) == "function") mm = mm()
mm$latent$rw2$min.diff = NULL
assign("inla.models", mm, INLA:::inla.get.inlaEnv())
co2res = inla(co2 ~ sin12 + cos12 + sin6 + cos6 + f(dateWeekInt,
  model = "rw2", values = as.integer(allDays), prior = "pc.prec",
  param = c(0.001, 0.5), scale.model = FALSE), data = co2s,
  family = "gaussian", control.family = list(hyper = list(prec = list(prior = "pc.prec",
    param = c(1, 0.5)))), control.inla = list(strategy = "gaussian"),
  control.compute = list(config = TRUE), verbose = TRUE)
qCols = c("0.5quant", "0.025quant", "0.975quant")
1/sqrt(co2res$summary.hyperpar[, qCols])
```

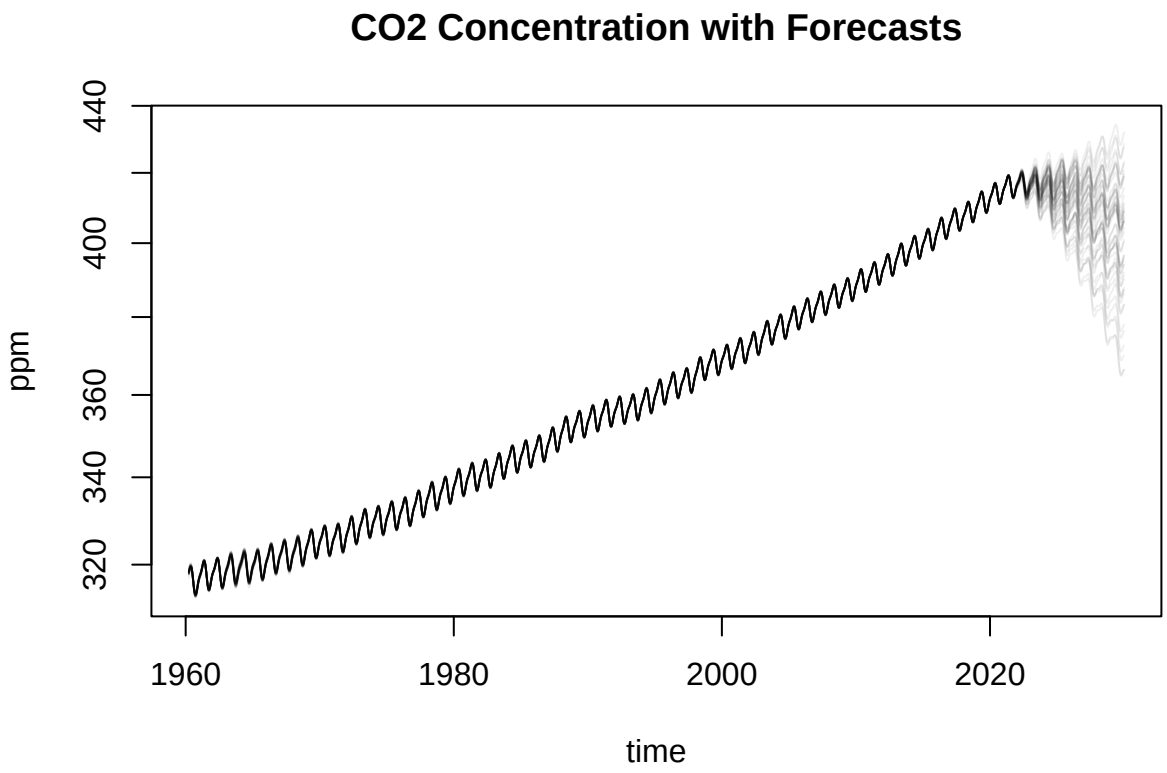
```
##                                             0.5quant   0.025quant   0.975quant
## Precision for the Gaussian observations 0.6078543957 0.6251706035 0.5910606757
## Precision for dateWeekInt               0.0001653206 0.0001937264 0.0001428522
```

```
matplot(co2res$summary.random$dateWeekInt[, qCols],
  type = "l", lty = 1)
```

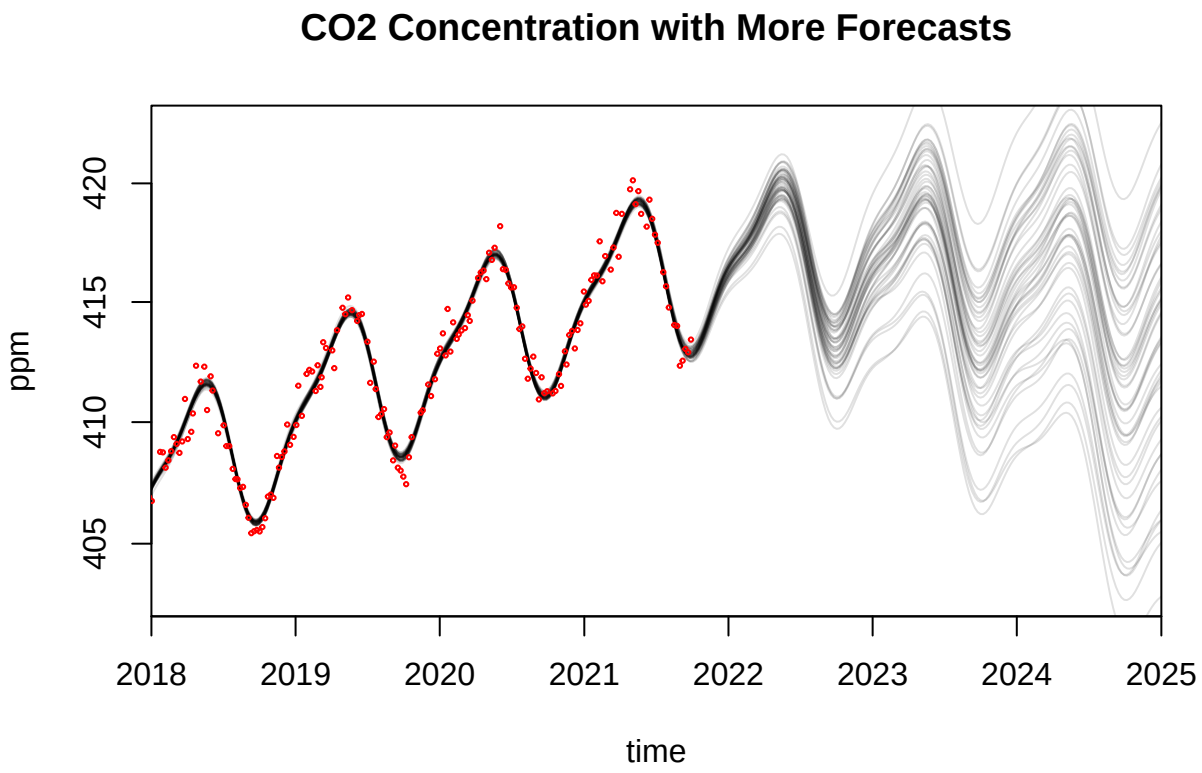



```
sampleList = INLA::inla.posterior.sample(50, co2res)
sampleMat = do.call(cbind, Biobase::subListExtract(sampleList,
  "latent"))
sampleMean = sampleMat[grep("dateWeekInt", rownames(sampleMat)),
  ]
sampleDeriv = apply(sampleMean, 2, diff) * (365.25/7)
forSinCos = 2 * pi * as.numeric(allDays)/365.25
forForecast = cbind(`(Intercept)` = 1, sin12 = sin(forSinCos),
  cos12 = cos(forSinCos), sin6 = sin(2 * forSinCos),
  cos6 = cos(2 * forSinCos))
forecastFixed = forForecast %*% sampleMat[paste0(colnames(forForecast),
  ":1"), ]
forecast = forecastFixed + sampleMean
```

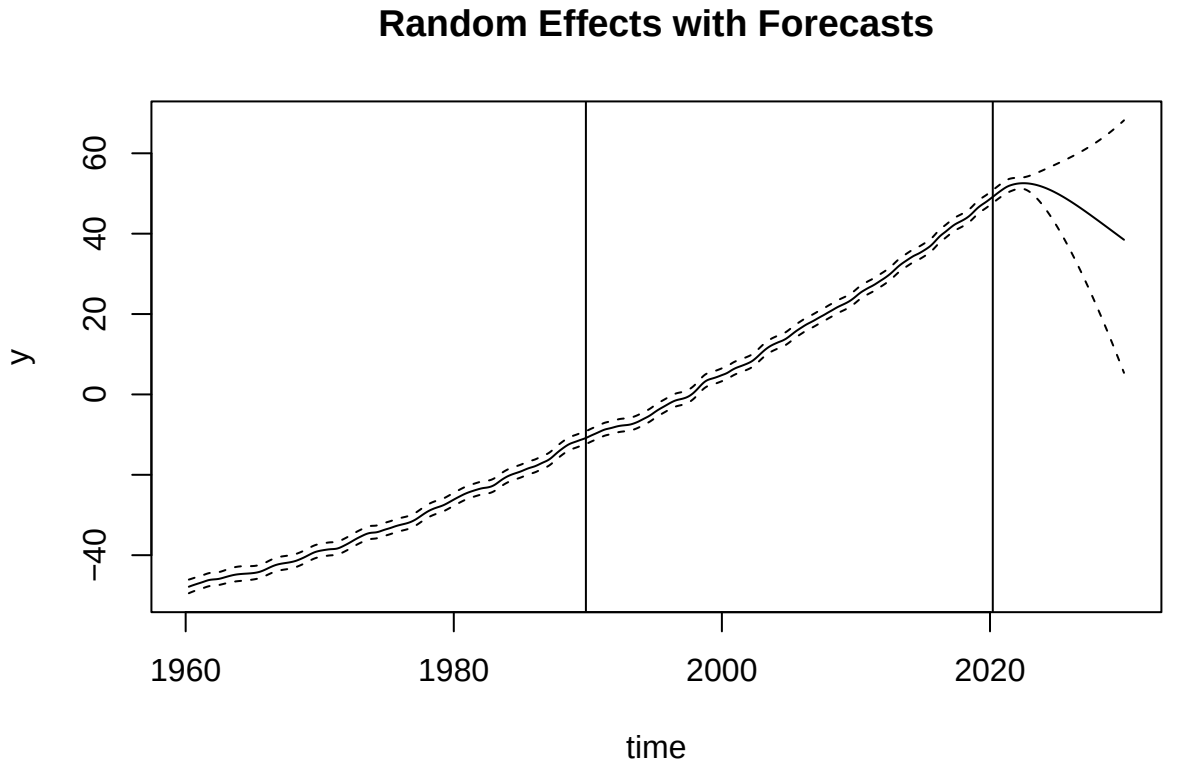
```
matplot(allDays, forecast, type = "l", col = "#00000010",
  lty = 1, log = "y", xlab = "time", ylab = "ppm", main="CO2 Concentration with Forecasts")
```



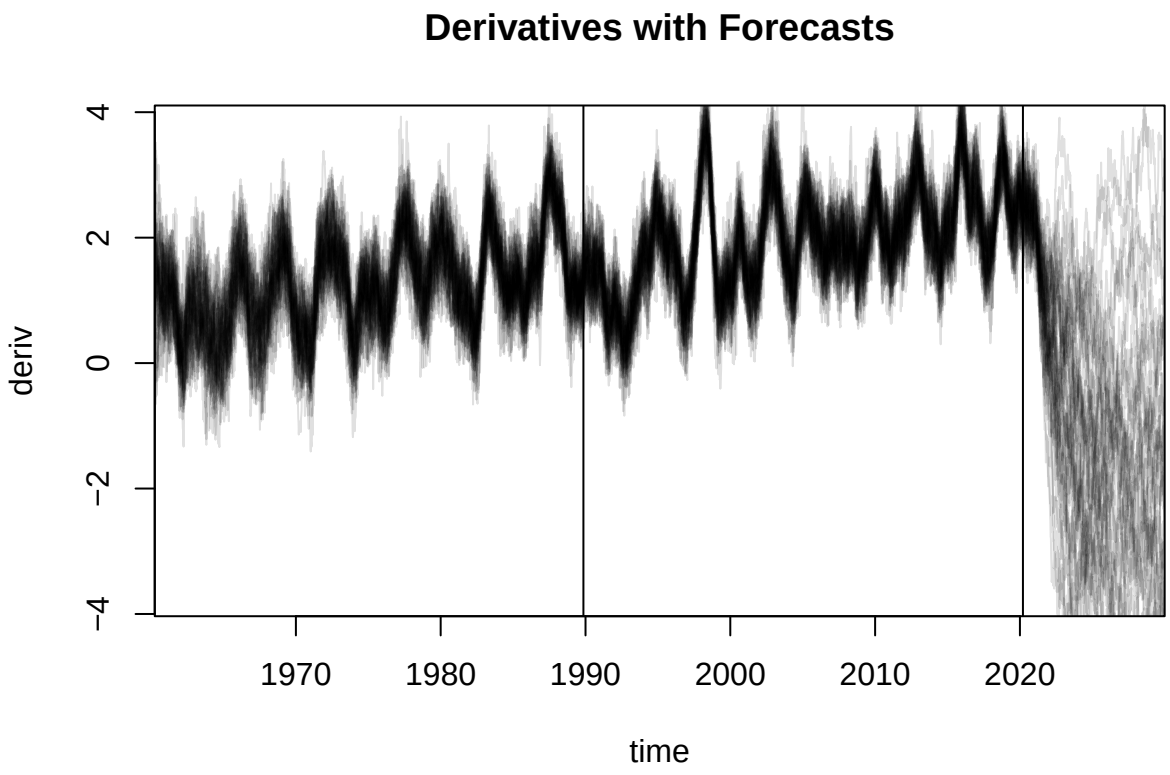

```
forX = as.Date(c("2018/1/1", "2025/1/1"))
forX = seq(forX[1], forX[2], by = "1 year")
toPlot = which(allDays > min(forX) & allDays < max(forX))
matplot(allDays, forecast, type = "l", col = "#00000020",
  lty = 1, log = "y", xlab = "time", ylab = "ppm",
  xaxs = "i", xaxt = "n", xlim = range(forX), ylim = range(forecast[which.min(abs(allDays -
    max(forX))), ]), main="CO2 Concentration with More Forecasts")
points(co2s$date, co2s$co2, col = "red", cex = 0.3)
axis(1, as.numeric(forX), format(forX, "%Y"))
```



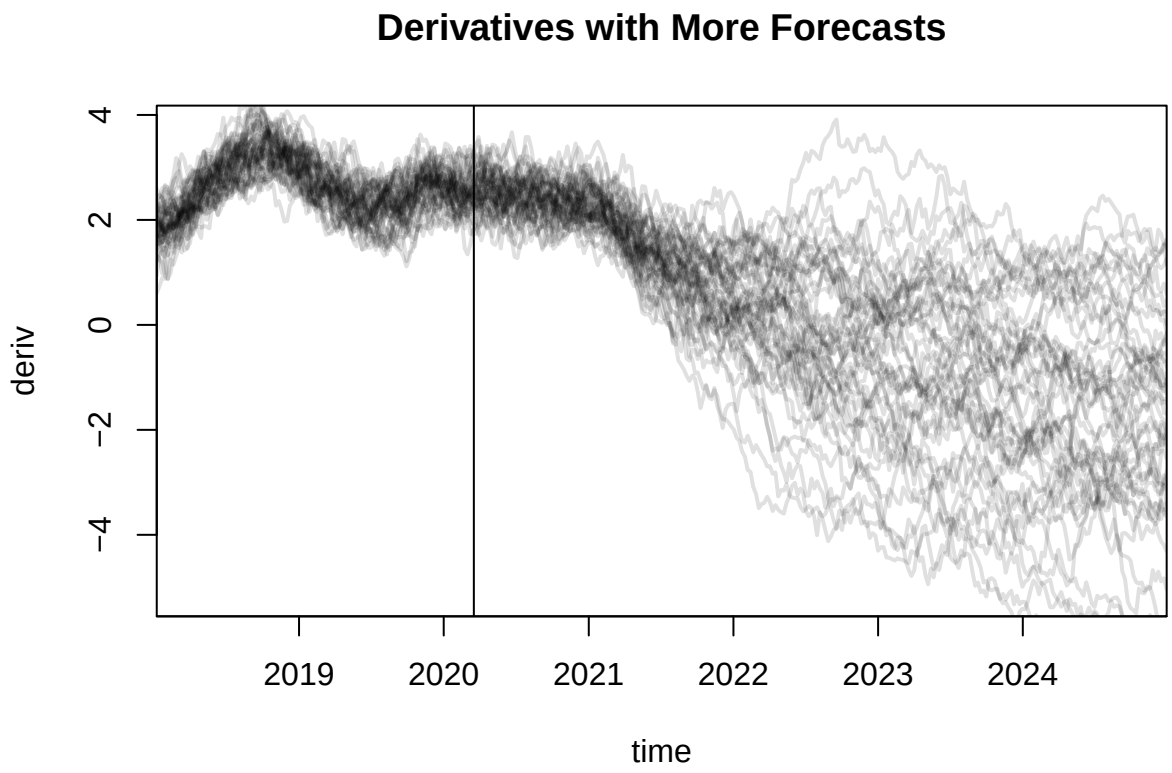

```
matplot(allDays, co2res$summary.random$dateWeekInt[,
  qCols], type = "l", col = "black", lty = c(1, 2,
  2), xlab = "time", ylab = "y", main="Random Effects with Forecasts")
abline(v=as.Date("1989/11/09"))
abline(v=as.Date("2020/03/17"))
```



```
matplot(allDays[-1], sampleDeriv, type = "l", lty = 1,
  xaxs = "i", col = "#00000020", xlab = "time", ylab = "deriv",
  ylim = quantile(sampleDeriv, c(0.025, 0.995)), main="Derivatives with Forecasts")
abline(v=as.Date("1989/11/09"))
abline(v=as.Date("2020/03/17"))
```




```
matplot(allDays[toPlot], sampleDeriv[toPlot, ], type = "l",
  lty = 1, lwd = 2, xaxs = "i", col = "#00000020",
  xlab = "time", ylab = "deriv", xaxt = "n", ylim = quantile(sampleDeriv[toPlot,
    ], c(0.01, 0.995)), main="Derivatives with More Forecasts")
axis(1, as.numeric(forX), format(forX, "%Y"))
abline(v=as.Date("2020/03/17"))
```


Derivatives with More Forecasts
deriv
4
2
0
−2
−4
2019
2020
2021
2022
2023
2024
time